## REMARKS/ARGUMENTS

Applicants respectfully request reconsideration of the present Application in view of the foregoing amendments and the following remarks/arguments. Claims 1-71 were originally filed with the present Application. By prior Amendment, claims 1 and 43 were amended, and claims 22-42 were canceled without prejudice or disclaimer in response to a Restriction Requirement. By this Amendment, claims 1 and 43 are again amended, and no new matter has been added. Accordingly, claims 1-21 and 43-71, as amended, are pending in the present Application.

## I. REJECTIONS UNDER 35 U.S.C. §103

The Examiner has rejected claims 1-21 and 43-71 under 35 U.S.C. §103(a) as allegedly obvious over U.S. Patent No. 6,477,580 to Bowman in view of U.S. Patent No. 6,665,861 to Francis, *et al.*, and further in view of U.S. Patent No. 6,438,559 to White, *et al.* In the present Amendment, Applicants have amended independent claims 1 and 43 to further clarify the invention(s) originally disclosed and claimed in the present Application.

In response to the above rejections, Applicants respectfully assert that independent claims 1 and 43, as amended, are not obvious in view of the cited combination of references for a number of reasons. In the prior Amendment, Applicants amended independent claims 1 and 43 to more clearly recite that the tagged data object for storing data is a *universal* tagged data object that is platform independent, hardware architecture independent, and language independent, and that holds data to provide universal access to manipulation and aggregation of the tagged data therein. The Examiner has again argued that Bowman discloses the claimed universal tagged data object by pointing to Bowman's use of an Internet browser. Specifically, the Examiner states that a "browser interpreting pages is equivalent to tagged data being stored in message or packages streamed between two browser applications." However, Applicants again respectfully disagree with the Examiner's assertion because browser applications do not and cannot exchange

data objects of the type recited in claims 1 and 43, and which are universal and thus platform, architecture, and language independent.

Specifically, the universal tagged data objects recited in claims 1 and 43 are represented in a binary format immediately capable of being processed by any processor on any platform without intermediate format conversion. In stark contrast, browsers employ HTML and XML formats to processes data they receive, both of which are textual formats and are not binary representations of data suitable for immediate processing. More specifically, a browser receives HTML and XML data as a serialized data transmission in a textual format. Then, in order to process the received data, the data that is in textual format, e.g., HTML, XML, etc., is converted to a binary representation (i.e., 1's and 0's) that can be processed by a processor running a browser. Once any processing is completed, the resulting data must be converted back to a textual format that may be read by the browser. In the recited claims, the tagged data is already in the binary format capable of being processed or directly transmitted to another computer. The receiving computer, which is running the API recited in claims 1 and 43 and that is configured to recognize the binary format being transmitted, then receives the transmitted data already in binary format. The data may then be processed by this receiving computer without an intermediate format conversion.

The elimination of the intermediate formatting/conversion steps has at least two significant advantages over transmitting textual data, such as with HTML and XML. First, the conversion of binary data to a text-based format and back again is a processor-intensive step that adds substantial processor overhead to an operation. Second, the conversion of binary data to a text-based format also imparts a substantial overhead to the size of the data. This can result in an unnecessary increase of transmission or processing time. As a result, Bowman does not disclose

or even suggest the use of a tagged data object for carrying transmitted data, where the tagged data object is represented in a universal binary format and transmitted in that format.

The Examiner, while admitting Bowman does not disclose a tagged data object that is universal and platform, architecture, and language independent, then argues that these universal characteristics of the tagged data object are "strongly suggested" by Bowman's use of COM format data and "language neutral" Java byte codes. However, Applicants respectfully assert that the Examiner's own cited examples of COM and Java code explicitly demonstrate how Bowman does not suggest, and actually teaches away from, any such universal and independent characteristics. Specifically, COM format is employed by Microsoft's Windows® platform and thus is not universal to other platforms. Similarly, the use of Java byte codes does not suggest the type of universal independence claimed in the present application since Java byte code is not used by non-Java-based systems and programs. As a result, although platform broker services may be capable of configuring some systems to exchange Java byte codes, architectures employing other languages, such as C++, are not compatible with Java codes. Moreover, the mere fact that a broker service has to be brought in to create some level of exchange uniformity is explicit evidence that Java byte codes are not universal and are not platform, architecture, and language independent, as are the universal tagged data objects recited in claims 1 and 43. Likewise, the use of ORB services among systems as cited by the Examiner may be somewhat possible, but such interchangeable capabilities are again provided by a broker and not by the independence of the data format. For example, if one endpoint uses Sun Systems® while another system uses HP® Systems, the use of ORB services in both systems does not suddenly make the two system completely compatible.

In addition, the Examiner has cited Francis for support of the use of binary representation of data that is universal, as recited in claims 1 and 43. In support of this assertion, the Examiner cites Francis' use of "Java beans with supporting utilities/metadata under markup or tagged form like XML." As discussed above, the XML format is a textual format, not the universal binary representation of data recited in claims 1 and 43. Even if 'Java beans' or other types of data are tagged in the data blocks of XML code, and the XML is serialized for transmission to another computer system, the serialized XML format is still not universal among all platforms, architectures, and languages. Serialized or deserialized, it is still a textual code understood using a specific language. Furthermore, even if the flexibility available with some textual languages like XML is used to create customized tags in the code for various types of data in an attempt to make the data universally employable among different systems, the customized code would now only be recognized by the system that generated the customized tags in the code, even if the data itself is in a readable format. As a result, with the customization allowed by such textual languages comes a decrease in its universal acceptance with other systems.

In addition, in the XML format example, when the data is to be transmitted, the data must be converted from XML format to serialized data, and then converted from the serialized data back to XML format at the receiving system. Then, the XML data must be converted to the machine-level binary representation of the data recited by the pending claims so that it may be processed within the receiving computer. The conversion from XML format to serialized data for transmission, and then after reception back to XML and then to machine-level binary data is time-consuming and unnecessarily consumes system resources. It is this intermediate conversion from the serialized data used during transmission back to XML (or other textual) format, before being converted into the binary data needed for processing, that is advantageously avoided by the

AMENDMENT UNDER 37 C.F.R. 1.113                                        PAGE 18 OF 20
DALDMS/507748.1

pending claims. Thus, the API recited in claims 1 and 43 is configured to aid in transmitting the raw binary data, as well as to recognize the transmitted binary data at the receiving system, without converting the data to a platform-, architecture-, language-, or program-specific textual format before processing it. As a result, Francis merely teaches the typical transmission of textual (i.e., non-binary) data in a specific manner, and does not teach or suggest the transmission of a machine-level binary representation of the tagged data as recited in claims 1 and 43. Thus, Francis does not cure the deficiencies of Bowman in this respect.

Finally, the Examiner relies on White for the suggestion to provide tagging of data object contents. Applicants also respectfully assert that the Examiner's reliance on White to teach or suggest the tag ID's recited in claims 1 and 43 is misplaced. As stated by the Examiner, White teaches "a method to serialize objects for distribution over a communications network environment using descriptors in serialization of class objects." To this end, White teaches the tagging of the object content being serialized with an identifier or value for serializing and deserializing data. However, White does not teach or suggest the use of binary tags to mark binary representations of the transmitted data. Instead, White simply discusses the typical tagging of data as part of a textual format, such as in XML code. Moreover, even if it can be said that White does disclose data tagging specifically as recited in claims 1 and 43, White does not cure the deficiencies of either Bowman or Francis discussed in detail above, and White is not relied on for such teachings.

For the reasons set forth in the prior Amendment, as well as the several reasons set forth above, the combinations of Bowman and Frances, or Bowman, Francis and White, do not teach or suggest all of the elements recited in independent claims 1 and 43. Since dependent claims 2-21 and 44-71 depend from independent claims 1 and 43, respectively, these claims are also not
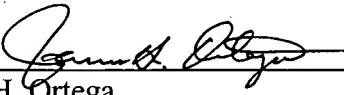
obvious in view of the cited references. Accordingly, Applicants respectfully request that the Examiner withdraw the §103(a) rejection with respect to claims 1-21 and 43-71.

## II.  CONCLUSION

Applicants respectfully submit that pending claims 1-21 and 43-71 are in condition for allowance, and request a Notice of Allowability for the pending claims. The Examiner is invited to contact the undersigned Attorney of Record if such would expedite the prosecution of the present Application. This Amendment and RCE is being filed within two months of the filing of the Notice of Appeal, which included a two month extension, and therefore no further fees are believed owed at this time. If it is determined that additional fees are due, or an overpayment has occurred, please charge or credit Deposit Account No. 13-0480, referencing Attorney Docket Number 68156755.5007.

Respectfully submitted,

Date: <u>December 3, 2004</u>

James H. Ortega
Reg. No. 50,554
Attorney for Applicants

**BAKER & McKENZIE LLP**
2300 Trammell Crow Center
2001 Ross Avenue
Dallas, TX 75201
Tel: (214) 978-3058
Fax: (214) 978-3099